CLAIMS

What is claimed is:

1. A method for coordinating the writing of data items to persistent storage, the method comprising the steps of:

maintaining within a first node a first queue for dirty data items that need to be written to persistent storage;

maintaining within the first node a second queue for dirty data items that need to be written to persistent storage;

moving entries from said first queue to said second queue when the dirty data items corresponding to the entries need to be transferred to a node other than said first node; and

when selecting which data items to write to persistent storage, given priority to data items that correspond to entries in said second queue.

2. The method of Claim 1 wherein the step of moving entries includes moving an entry from said first queue to said second queue in response to a message received by said first node, wherein said message indicates that another node has requested the data item that corresponds to said entry.

3. A method for coordinating the writing of data items to persistent storage, the method comprising the steps of:

maintaining a forced-write count for each of said data items;

incrementing the forced-write count of a data item whenever the data item is written to persistent storage by one node for transfer of the data item to another node;

selecting which dirty data items to write to persistent storage based on the write counts associated with the data items.

4. The method of Claim 3 further comprising the steps of:

storing dirty data items that have forced-write counts above a certain threshold in a particular queue; and

when selecting dirty data items to write to persistent storage, giving priority to data items stored in said particular queue.

5. A method of managing information about where to begin recovery after a failure, the method comprising the steps of:

in a particular node of a multiple-node system, maintaining both

a single-failure queue that indicates where within a recovery log to begin recovery after a failure of said node, and

a multiple-failure queue that indicates where within said recovery log to begin recovery after a failure of said node and one or more other nodes in said multiple-node system;

in response to a dirty data item being written to persistent storage, removing an entry for said data item from both said single-failure queue and said multiple-failure queue; and

in response to a dirty data item being sent to another node of said multiple-node system without first being written to persistent storage, removing an entry for said data item from said single-failure queue without removing the entry for said data item from said multiple-failure queue.

6. The method of Claim 5 further comprising the step of sending the dirty data item to another node to allow removal of the entry from said single-failure queue without the other node requesting the dirty data item.

7. The method of Claim 5 further comprising the steps of:

after a single node failure, applying said recovery log beginning at a position in said recovery log associated with the single-failure queue; and

after a multiple node failure, applying said recovery log beginning at a position in said recovery log associated with the multiple-failure queue.

8. The method of Claim 5 wherein:

said single-failure queue and said multiple-failure queue are implemented by a single combined queue; and

the step of removing an entry for said data item from said single-failure queue without removing the entry for said data item from said multiple-failure queue includes marking an entry for said data item in said combined queue without removing the entry for said data item from said combined queue.

9. The method of Claim 5 wherein said single-failure queue and said multiple-failure queue are implemented as two separate queues.

10. A method for recovering after a failure, the method comprising the steps of:

determining whether the failure involves only one node; and

if the failure involves only said one node, then performing recovery by applying a recovery log of said node beginning at a first point in the recovery log; and

if the failure involves one or more nodes in addition to said one node, then performing recovery by applying said recovery log of said node beginning at a second point in the recovery log;

wherein said first point is different from said second point.

11. The method of Claim 10 wherein:

the first point is determined, at least in part, by which data items that were dirtied by said node reside in caches in other nodes; and

the second point is determined, at least in part, by which data items that were dirtied by said node have been persistently stored.

12. A method for recovering after a failure, the method comprising the steps of:

if it is unclear whether a particular version of a data item has been written to disk, then performing the steps of

without attempting to recover said data item, marking dirtied cached versions of said data item that would have been covered if said particular version was written to disk;

when a request is made to write one of said dirtied cached versions to disk, determining which version of said data item is already on disk; and

if said particular version of said data item is already on disk, then not writing said one of said dirtied cached versions to disk.

13. The method of Claim 12 further comprising the step of, if said particular version of said data item is not already on disk, then recovering said data item.

14. The method of Claim 12 further comprising the step of, if said particular version of said data item is already on disk, then informing nodes that contain said dirtied cached versions of the data item that said dirtied cached versions are covered by a write-to-disk operation.

15. A method for recovering a current version of a data item after a failure in a system that includes multiple caches, the method comprising the steps of:

modifying the data item in a first node of said multiple caches to create a modified data item;

sending the modified data item from said first node to a second node of said multiple caches without durably storing the modified data item from said first node to persistent storage;

after said modified data item has been sent from said first node to said second node and before said data item in said first node has been covered by a write-to-disk operation, discarding said data item in said first node; and

after said failure, reconstructing the current version of said data item by applying changes to the data item on persistent storage based on merged redo logs associated with all of said multiple caches.

16. The method of Claim 15 further comprising the steps of:

maintaining, for each of said multiple caches, a globally-dirty checkpoint queue and a locally-dirty checkpoint queue;

wherein the globally-dirty data items associated with entries in the globally-dirty checkpoint queue are not retained until covered by write-to-disk operations;

determining, for each cache, a checkpoint based on a lower of a first-dirtied time of the entry at the head of the locally-dirty checkpoint queue and the first-dirtied time of the entry at the head of the globally-dirty checkpoint queue; and

after said failure, determining where to begin processing the redo log associated with each cache based on the checkpoint determined for said cache.

17. The method of Claim 15 further comprising the steps of:

maintaining, for each of said multiple caches, a globally-dirty checkpoint queue and a locally-dirty checkpoint queue;

wherein the globally-dirty data items associated with entries in the globally-dirty checkpoint queue are not retained until covered by write-to-disk operations;

maintaining, for each cache, a first checkpoint record for the locally-dirty checkpoint queue that indicates a first time, where all changes made to data items that are presently dirty in the cache prior to the first time have been recorded on a version of the data item that is on persistent storage;

maintaining, for each cache, a second checkpoint record for the globally-dirty checkpoint queue, wherein the second checkpoint record includes a list of data items that were once dirtied in the cache but have since been transferred out and not written to persistent storage; and

after said failure, determining where to begin processing the redo log associated with each cache based on the first checkpoint record and said second checkpoint record for said cache.

18. The method of Claim 17 wherein the step of processing the redo log comprises the steps of:

determining a starting position for scanning the redo log based on a lesser of a position in the redo log as determined by the first checkpoint record and the positions in the log as determined by the earliest change made to the list of the data items in the second checkpoint record; and

during recovery, for the portion of the redo log between the position indicated by the global checkpoint record to the position indicated by the local checkpoint record, considering for potential redo only those log records that correspond to the data items identified in the global checkpoint record.

19. A computer-readable medium carrying instructions for coordinating the writing of data items to persistent storage, the instructions comprising instructions for performing the steps of:

maintaining within a first node a first queue for dirty data items that need to be written to persistent storage;

maintaining within the first node a second queue for dirty data items that need to be written to persistent storage;

moving entries from said first queue to said second queue when the dirty data items corresponding to the entries need to be transferred to a node other than said first node; and

when selecting which data items to write to persistent storage, given priority to data items that correspond to entries in said second queue.

20. The computer-readable medium of Claim 19 wherein the step of moving entries includes moving an entry from said first queue to said second queue in response to a message received by said first node, wherein said message indicates that another node has requested the data item that corresponds to said entry.

21. A computer-readable medium carrying instructions for coordinating the writing of data items to persistent storage, the instructions comprising instructions for performing the steps of:

maintaining a forced-write count for each of said data items;

incrementing the forced-write count of a data item whenever the data item is written to persistent storage by one node for transfer of the data item to another node;

selecting which dirty data items to write to persistent storage based on the write counts associated with the data items.

22. The computer-readable medium of Claim 21 further comprising instructions for performing the steps of:

storing dirty data items that have forced-write counts above a certain threshold in a particular queue; and

when selecting dirty data items to write to persistent storage, giving priority to data items stored in said particular queue.

23. A computer-readable medium carrying instructions for managing information about where to begin recovery after a failure, the instructions comprising instructions for performing the steps of:

in a particular node of a multiple-node system, maintaining both

a single-failure queue that indicates where within a recovery log to begin recovery after a failure of said node, and

a multiple-failure queue that indicates where within said recovery log to begin recovery after a failure of said node and one or more other nodes in said multiple-node system;

in response to a dirty data item being written to persistent storage, removing an entry for said data item from both said single-failure queue and said multiple-failure queue; and

in response to a dirty data item being sent to another node of said multiple-node system without first being written to persistent storage, removing an entry for said data item from said single-failure queue without removing the entry for said data item from said multiple-failure queue.

24. The computer-readable medium of Claim 23 further comprising instructions for performing the step of sending the dirty data item to another node to allow removal of the entry from said single-failure queue without the other node requesting the dirty data item.

25. The computer-readable medium of Claim 23 further comprising instructions for performing the steps of:

after a single node failure, applying said recovery log beginning at a position in said recovery log associated with the single-failure queue; and

after a multiple node failure, applying said recovery log beginning at a position in said recovery log associated with the multiple-failure queue.

26. The computer-readable medium of Claim 23 wherein:

said single-failure queue and said multiple-failure queue are implemented by a single combined queue; and

the step of removing an entry for said data item from said single-failure queue without removing the entry for said data item from said multiple-failure queue includes marking an entry for said data item in said combined queue without removing the entry for said data item from said combined queue.

27. The computer-readable medium of Claim 23 wherein said single-failure queue and said multiple-failure queue are implemented as two separate queues.

28. A computer-readable medium carrying instructions for recovering after a failure, the instructions comprising instructions for performing the steps of:

determining whether the failure involves only one node; and

if the failure involves only said one node, then performing recovery by applying a recovery log of said node beginning at a first point in the recovery log; and

if the failure involves one or more nodes in addition to said one node, then performing recovery by applying said recovery log of said node beginning at a second point in the recovery log;

wherein said first point is different from said second point.

29. The computer-readable medium of Claim 28 wherein:

the first point is determined, at least in part, by which data items that were dirtied by said node reside in caches in other nodes; and

the second point is determined, at least in part, by which data items that were dirtied by said node have been persistently stored.

30. A computer-readable medium carrying instructions for recovering after a failure, the instructions comprising instructions for performing the steps of:

if it is unclear whether a particular version of a data item has been written to disk, then performing the steps of

without attempting to recover said data item, marking dirtied cached versions of said data item that would have been covered if said particular version was written to disk;

when a request is made to write one of said dirtied cached versions to disk, determining which version of said data item is already on disk; and

if said particular version of said data item is already on disk, then not writing said one of said dirtied cached versions to disk.

31. The computer-readable medium of Claim 30 further comprising instructions for performing the step of, if said particular version of said data item is not already on disk, then recovering said data item.

32. The computer-readable medium of Claim 30 further comprising instructions for performing the step of, if said particular version of said data item is already on disk, then informing nodes that contain said dirtied cached versions of the data item that said dirtied cached versions are covered by a write-to-disk operation.

33. A computer-readable medium carrying instructions for recovering a current version of a data item after a failure in a system that includes multiple caches, the instructions comprising instructions for performing the steps of:

modifying the data item in a first node of said multiple caches to create a modified data item;

sending the modified data item from said first node to a second node of said multiple caches without durably storing the modified data item from said first node to persistent storage;

after said modified data item has been sent from said first node to said second node and before said data item in said first node has been covered by a write-to-disk operation, discarding said data item in said first node; and

after said failure, reconstructing the current version of said data item by applying changes to the data item on persistent storage based on merged redo logs associated with all of said multiple caches.

34. The computer-readable medium of Claim 33 further comprising instructions for performing the steps of:

maintaining, for each of said multiple caches, a globally-dirty checkpoint queue and a locally-dirty checkpoint queue;

wherein the globally-dirty data items associated with entries in the globally-dirty checkpoint queue are not retained until covered by write-to-disk operations;

determining, for each cache, a checkpoint based on a lower of a first-dirtied time of the entry at the head of the locally-dirty checkpoint queue and the first-dirtied time of the entry at the head of the globally-dirty checkpoint queue; and

after said failure, determining where to begin processing the redo log associated with each cache based on the checkpoint determined for said cache.

35. The computer-readable medium of Claim 33 further comprising instructions for performing the steps of:

maintaining, for each of said multiple caches, a globally-dirty checkpoint queue and a locally-dirty checkpoint queue;

wherein the globally-dirty data items associated with entries in the globally-dirty checkpoint queue are not retained until covered by write-to-disk operations;

maintaining, for each cache, a first checkpoint record for the locally-dirty checkpoint queue that indicates a first time, where all changes made to data items that are presently dirty in the cache prior to the first time have been recorded on a version of the data item that is on persistent storage;

maintaining, for each cache, a second checkpoint record for the globally-dirty checkpoint queue, wherein the second checkpoint record includes a list of data items that were once dirtied in the cache but have since been transferred out and not written to persistent storage; and

after said failure, determining where to begin processing the redo log associated with each cache based on the first checkpoint record and said second checkpoint record for said cache.

36. The computer-readable medium of Claim 35 wherein the step of processing the redo log comprises the steps of:

determining a starting position for scanning the redo log based on a lesser of

a position in the redo log as determined by the first checkpoint record and

the positions in the log as determined by the earliest change made to the list of the data items in the second checkpoint record; and

during recovery, for the portion of the redo log between the position indicated by the global checkpoint record to the position indicated by the local checkpoint

record, considering for potential redo only those log records that correspond to the data items identified in the global checkpoint record.